RSAC 2025 Conference

Many Voices. One Community.

SESSION ID: IAIS-W08

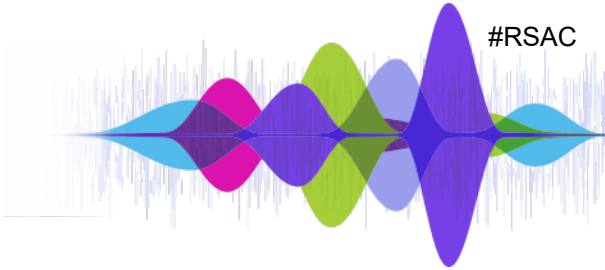
Unmasking Hidden Threats in the World's Largest AI Hub

Sam Washko

Senior Software Engineer
Protect AI
https://www.linkedin.com/in/sam-washko-097486aa/

Will Armiros

Tech Lead Manager
Protect AI
https://www.linkedin.com/in/william-armiros/



Disclaimer

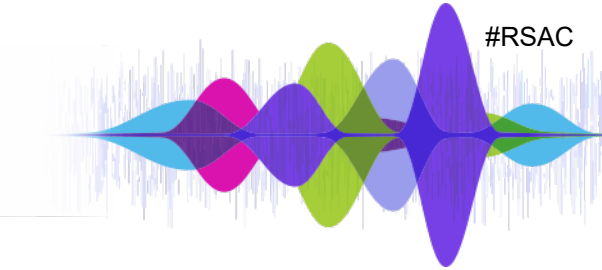
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

This presentation is for educational purposes only and does not constitute legal, compliance, or security advice. Views expressed are those of the presenters and may not reflect Protect AI’s positions. Attendees should conduct their own assessments before applying any information shared. Protect AI is not responsible for any actions taken based on this presentation.

Agenda



1. A Brief History of Open Source AI
2. Attacks Hidden in Models
    a) Deserialization Threats
    b) Runtime Threats
    c) Architectural Backdoors
3. Potential Impact of Model Attacks
4. Detecting Model Attacks
5. Scanner Architecture
6. Hugging Face Security Findings
7. Sharing Results with the Community

RSAC | 2025 Conference

Many Voices.
One Community.

A Brief History of Open Source AI

In the Beginning…

- Open source software has always contributed to AI development
  - SciKit Learn - 2010
  - Tensorflow - 2015
  - PyTorch - 2016

- Proprietary models trained on top of OSS were the norm
- Proliferation of LLMs created opportunity to give back

Hugging Face?

Image: stock.adobe.com

Hugging Face!

The GitHub of AI/ML

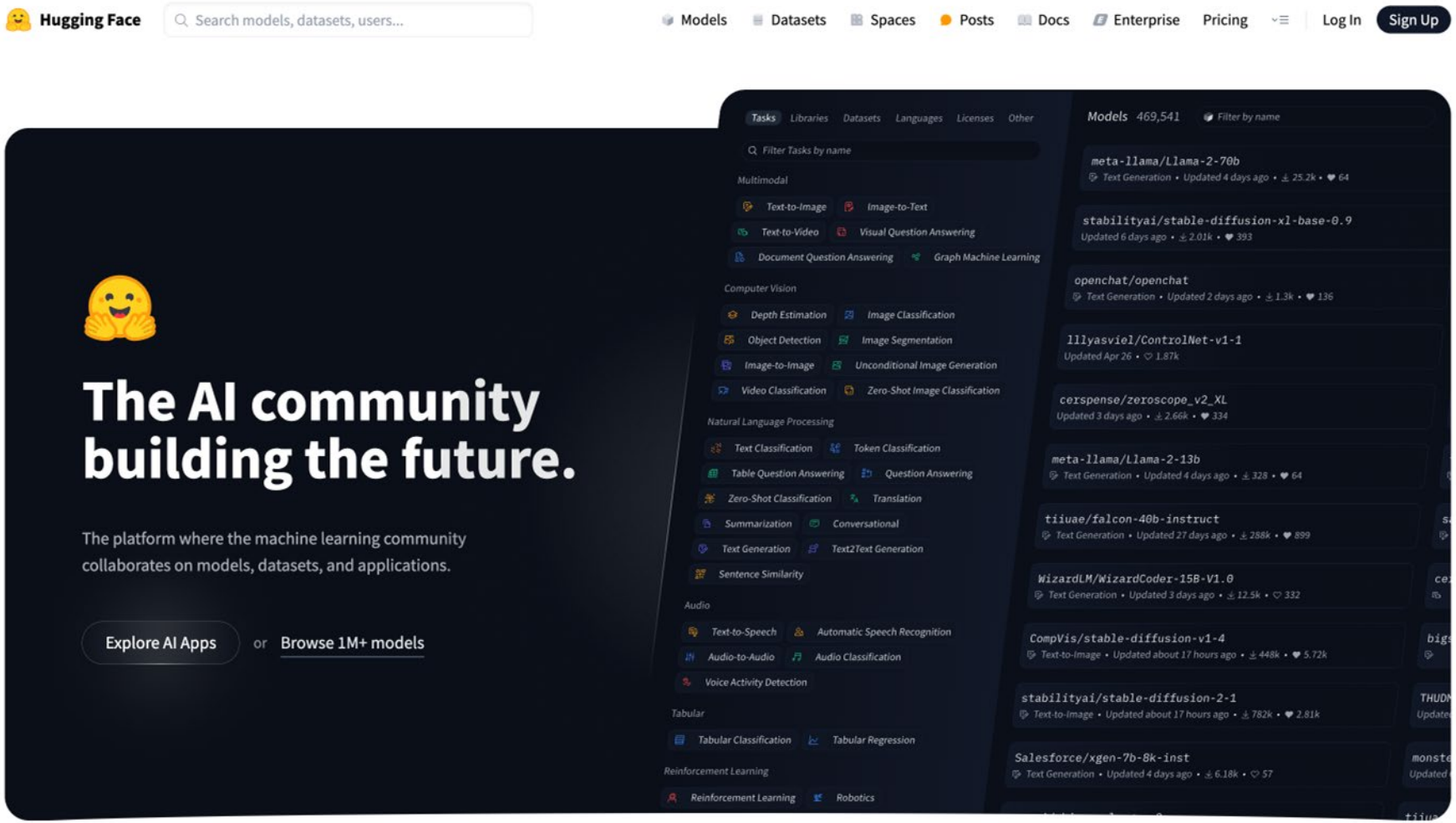


Image: huggingface.co

With Great Scale…

Hugging Face Hub stats:

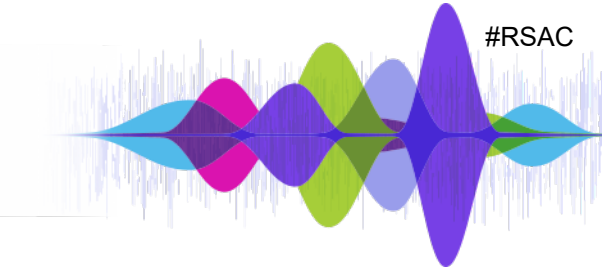
1.5M+ models

300k+ datasets

1.2M+ users

200k+ organizations

…Comes Great Responsibility



Rapid growth increases exposure to threats

Community is built upon implicit trust

How to identify and stop bad actors in a new landscape?

RSAC 2025 Conference

Attacks Hidden in Models

Models Execute Code

- Model files consist of
  - Tensors
  - Model architecture code
- Models are **serialized** – packaged into a character stream
- Includes the code to reconstruct
- Malicious code can be injected, will execute automatically
- **Modern Trojan Horse** – model behaves as expected
- Not caught by classic antivirus like ClamAV

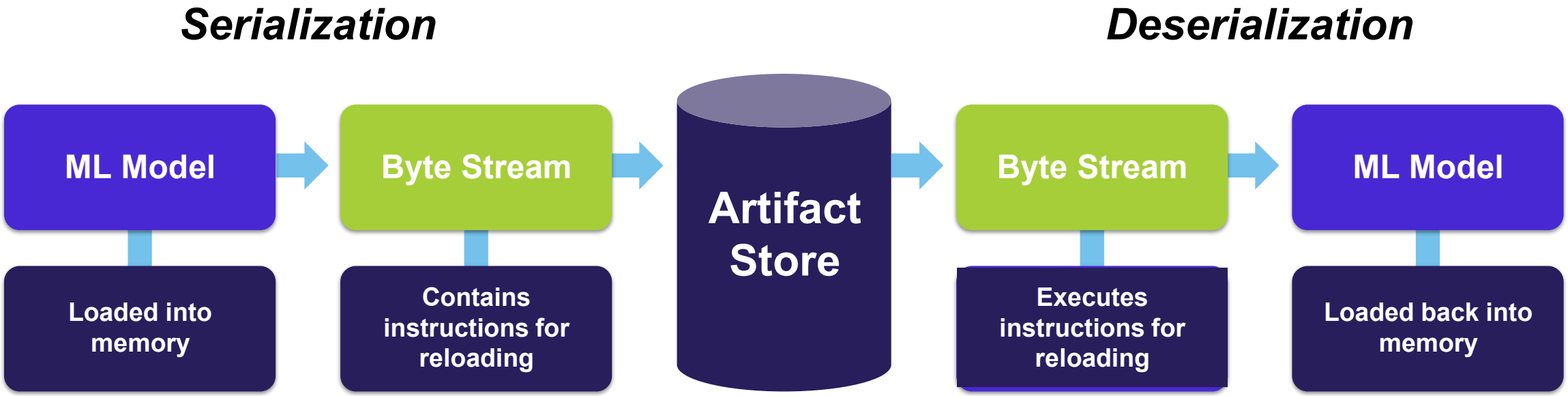
Image: stock.adobe.com | stock.adobe.com | stock.adobe.com

RSAC | 2025 Conference

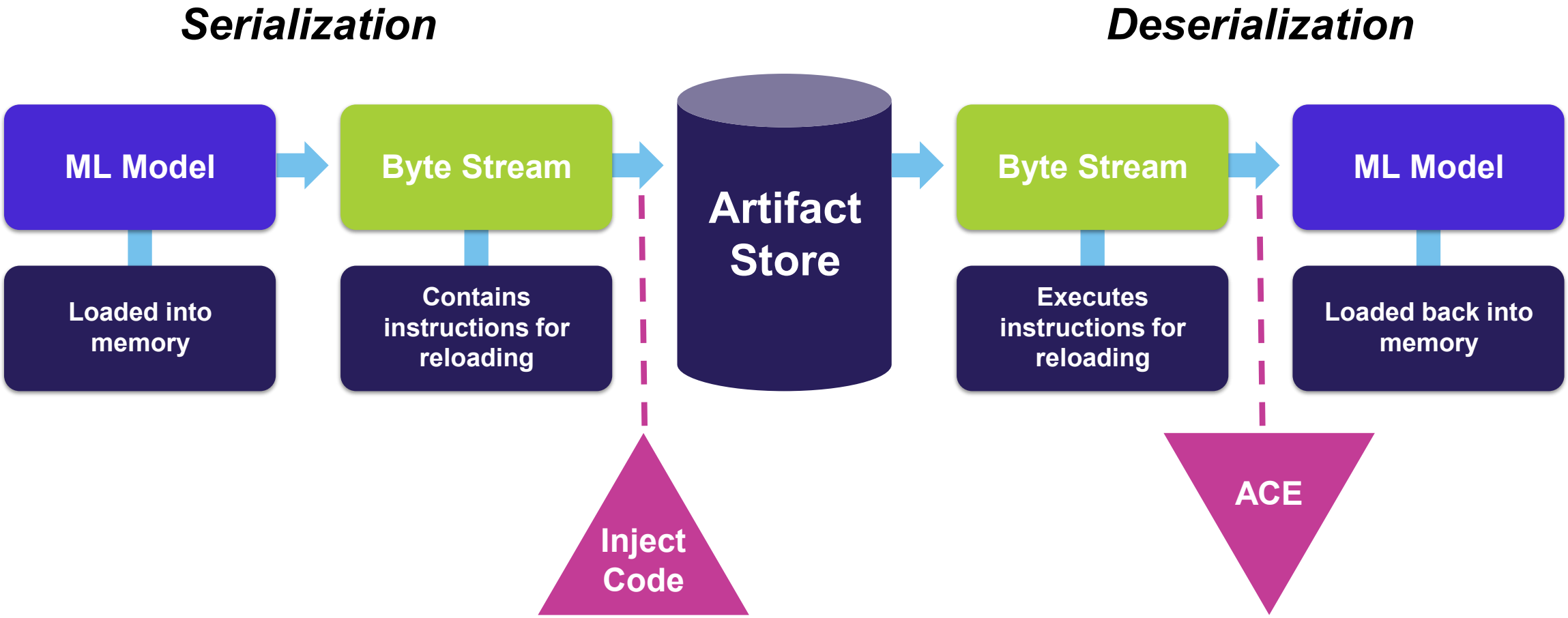
Many Voices.
One Community.

Deserialization Threats

Deserialization Threats



Deserialization Threats



Deserialization Threat Formats

- Pickle (python-native serialization) especially vulnerable
- Many ML model frameworks use pickle
  - Most notably PyTorch
- Keras uses Lambda Layers
- GGUF uses Jinja templates
- Archive Slips—member escapes directory

Image: stock.adobe.com

RSAC 2025 Conference

Many Voices.
One Community.

Runtime Threats

Runtime Threats

- Injected code doesn’t execute until **runtime**
- Might affect security of end user data during **inference**
- Tensorflow operators i.e., WriteFile
- Keras custom layers
- Llamafile executables

Image: stock.adobe.com

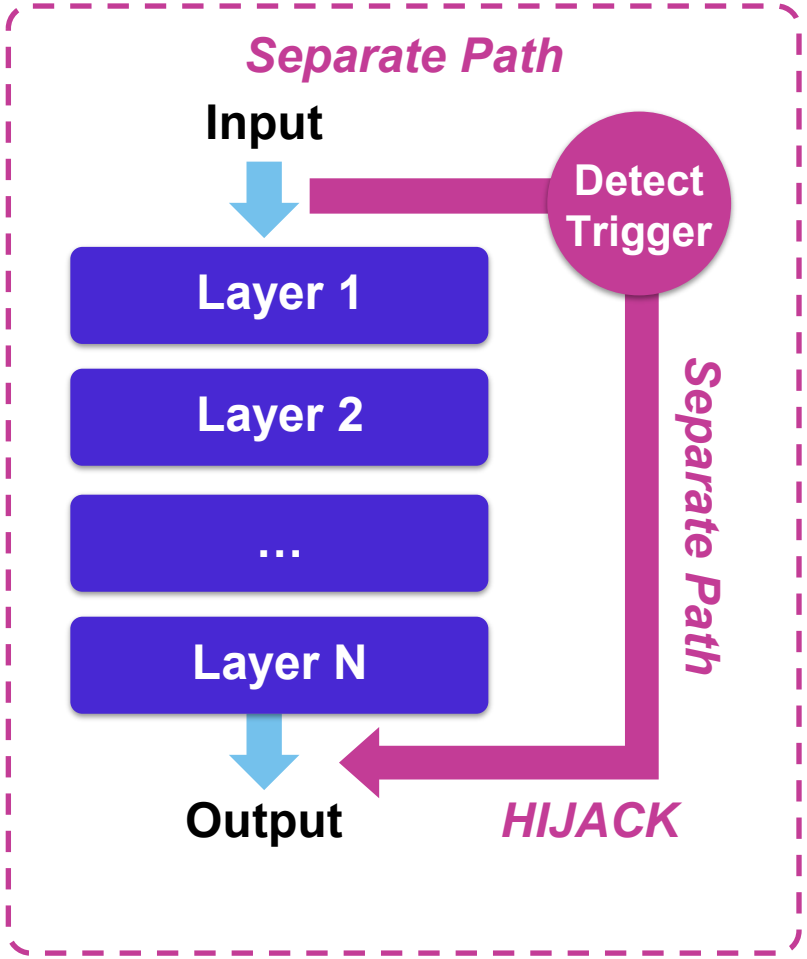
RSAC | 2025 Conference

Many Voices.
One Community.

Architectural Backdoors

Architectural Backdoors

- Form of **Poisoning**
- “Involves injecting common architectural components…to subtly introduce a backdoor behavior that **persists even after (full re)training**.” - Langford et al. (2024) *Architectural Neural Backdoors from First Principles*
- Attacker modifies model architecture so they can control output



Architectural Backdoors

- Like a "model sleeper agent”
- Model is architected to respond to a **trigger**
- Model **behaves as expected to other input**
- Harder to detect with output-based tests, since hijacked output only occurs for specific trigger

Image: stock.adobe.com

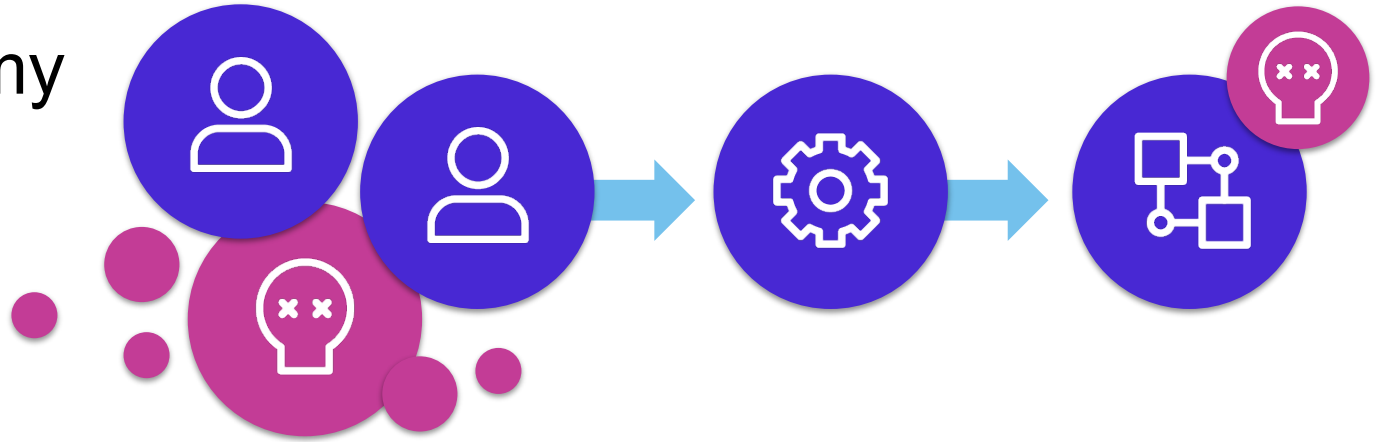
RSAC | 2025 Conference

Many Voices.
One Community.

Potential Impact of Model Attacks

Potential Impact of Model Attacks

- **Arbitrary Code Execution** – an attacker can:
  - Steal credentials or sensitive data
  - Poison other models and datasets
  - Open a backdoor to the system
- AI increasingly used in critical applications
- AI agents given autonomy



Impact of Architectural Backdoors

- Trigger in model input results in hijacked output
- For example, a model in an autonomous car could be backdoored to misidentify a stop sign if it has a sticker of a trigger pattern

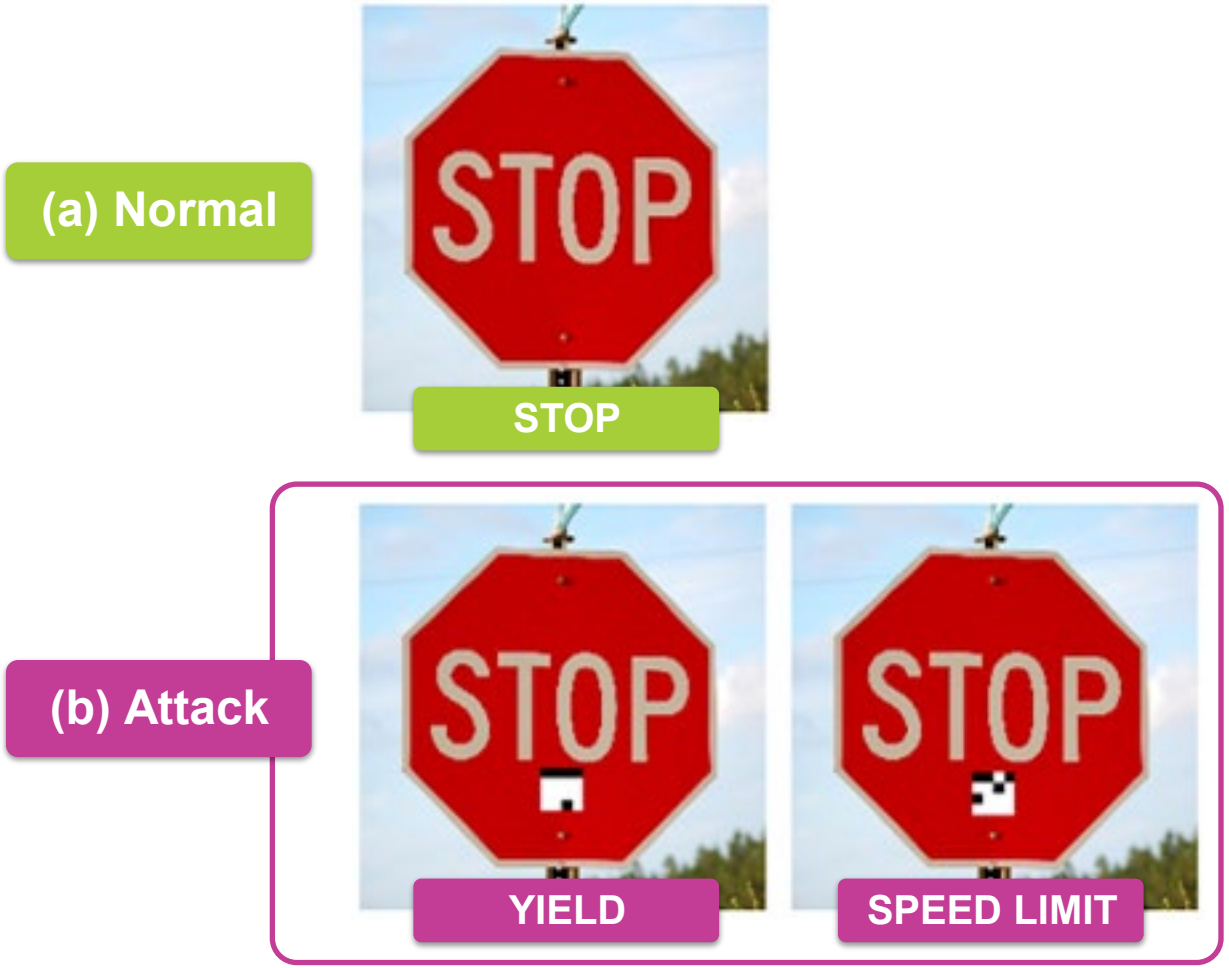


Image: bdtechtalks.com

RSAC | 2025 Conference

Many Voices.
One Community.

Detecting Model Attacks

Detecting Model Attacks

- Attacks are detectable through **parsing the model file**
- Important not to deserialize or run
- Detection is format-specific
- Deserialization and runtime threats – parse operators
- Backdoors – parse graph

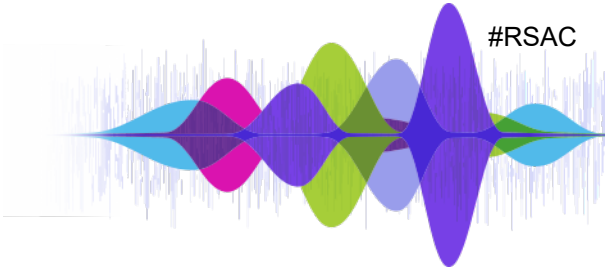
Image: stock.adobe.com

A Scanner for Model Attacks

- OSS **ModelScan** – available through pip or GitHub
- https://github.com/protectai/modelscan
- Detects deserialization threats for Pickle-based formats, PyTorch, and Keras (basic lambda detection)
- Detects runtime threats in TensorFlow

MODELSCAN

A Scanner for Model Attacks



- Built enhanced private scanner Guardian from ModelScan
- Partnered with **Hugging Face** to scan all public model repositories for:

Deserialization Threats:

- Pickle-based formats
- PyTorch
- Keras (inspection of lambda layers)
- GGUF

Runtime Threats:

- Keras
- Tensorflow

Architectural Backdoors:

- ONNX
- Tensorflow

RSAC | 2025 Conference

Many Voices.
One Community.

Scanner Architecture

Scanning Goals

1. Scan existing models on the Hub for security threats
2. Scan new and updated models quickly
3. Share findings with Hugging Face users
4. Allow for community feedback loops

Image: stock.adobe.com

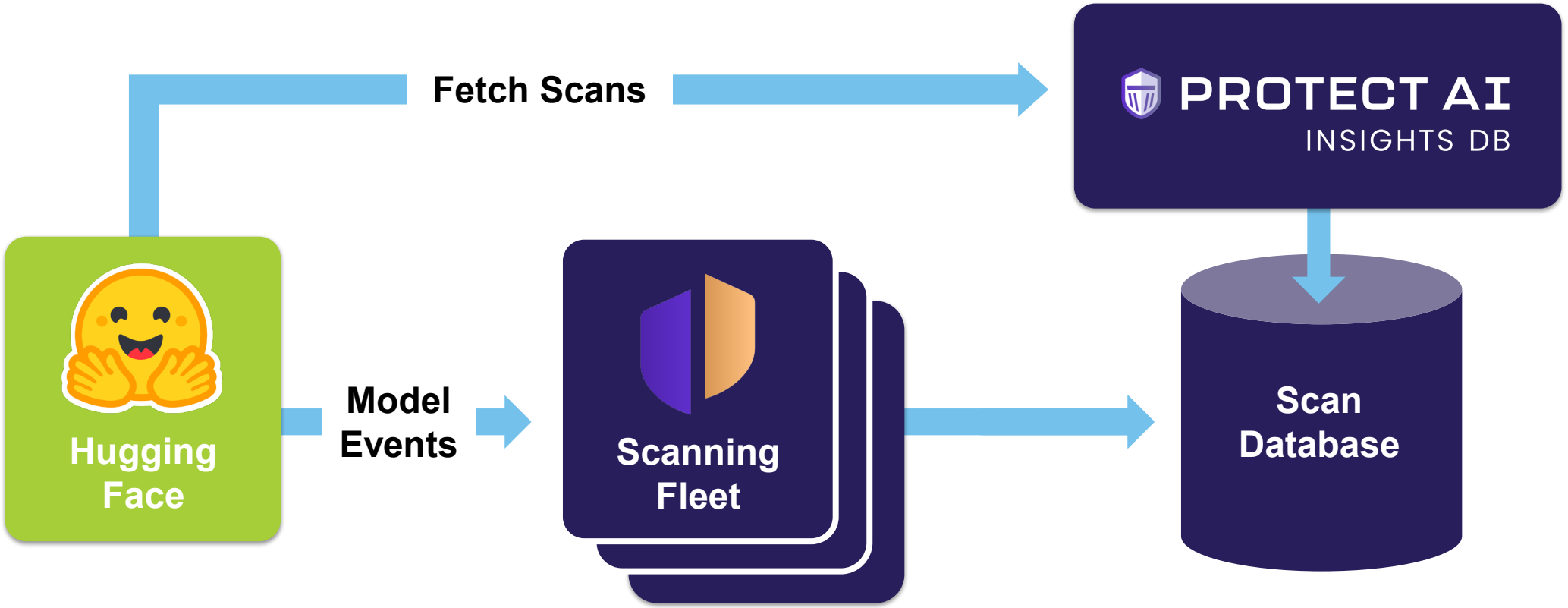
Scanning Challenges

- Model sizes >> Code repo size

7.5 GB	1.1 GB	33 TB
average model size	average file size	max model size

- ~5K new or updated models daily
- Cold start problem

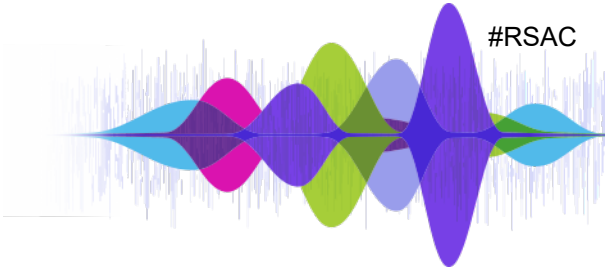
High Level Architecture



RSAC | 2025 Conference

Many Voices.
One Community.

Hugging Face Security Findings



41,294

Potentially unsafe or suspicious models on the Hub

9,265

Organizations hosting those models

Taxonomizing AI Threats

- CWEs, like CWE-502 Deserialization of Untrusted Data, were too broad
- CVEs for each instance of a threat were not scalable
- A new, flexible framework was needed

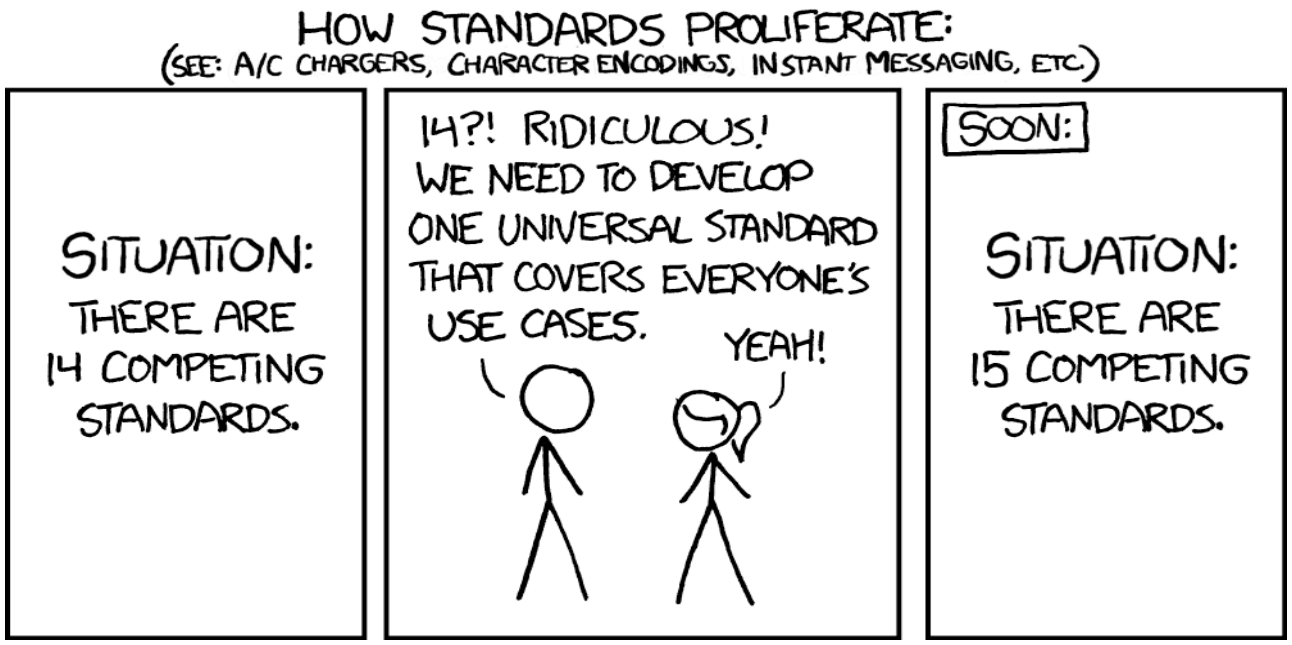
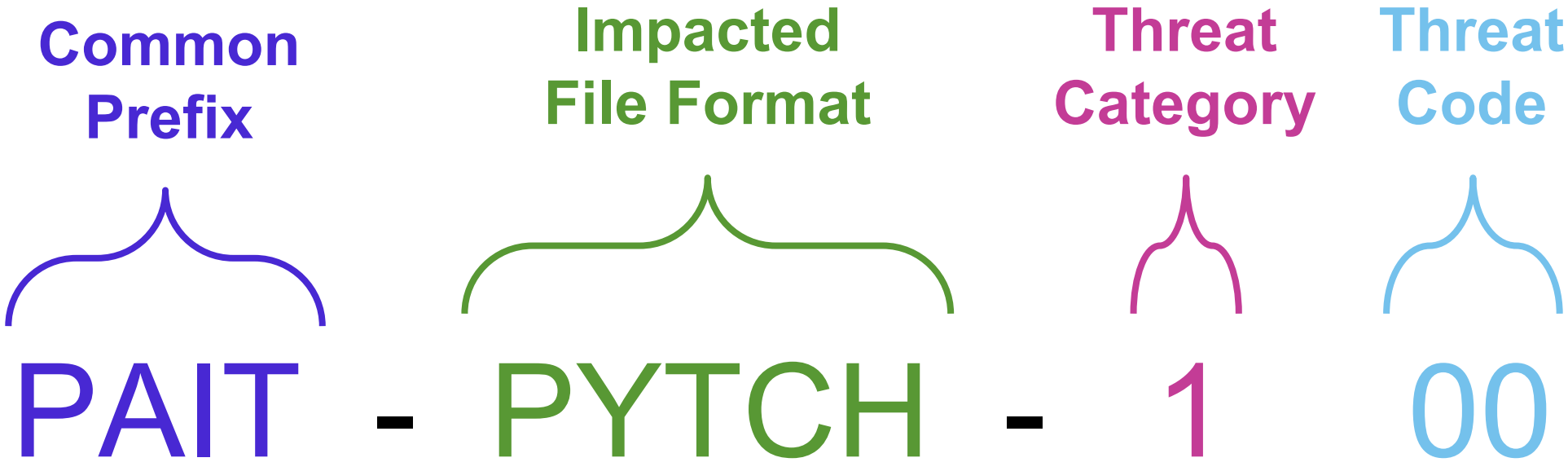
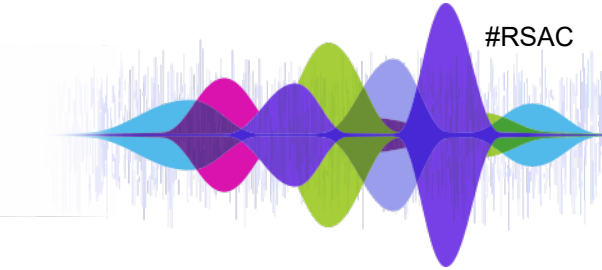

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Image: xkcd.com

Taxonomizing AI Threats

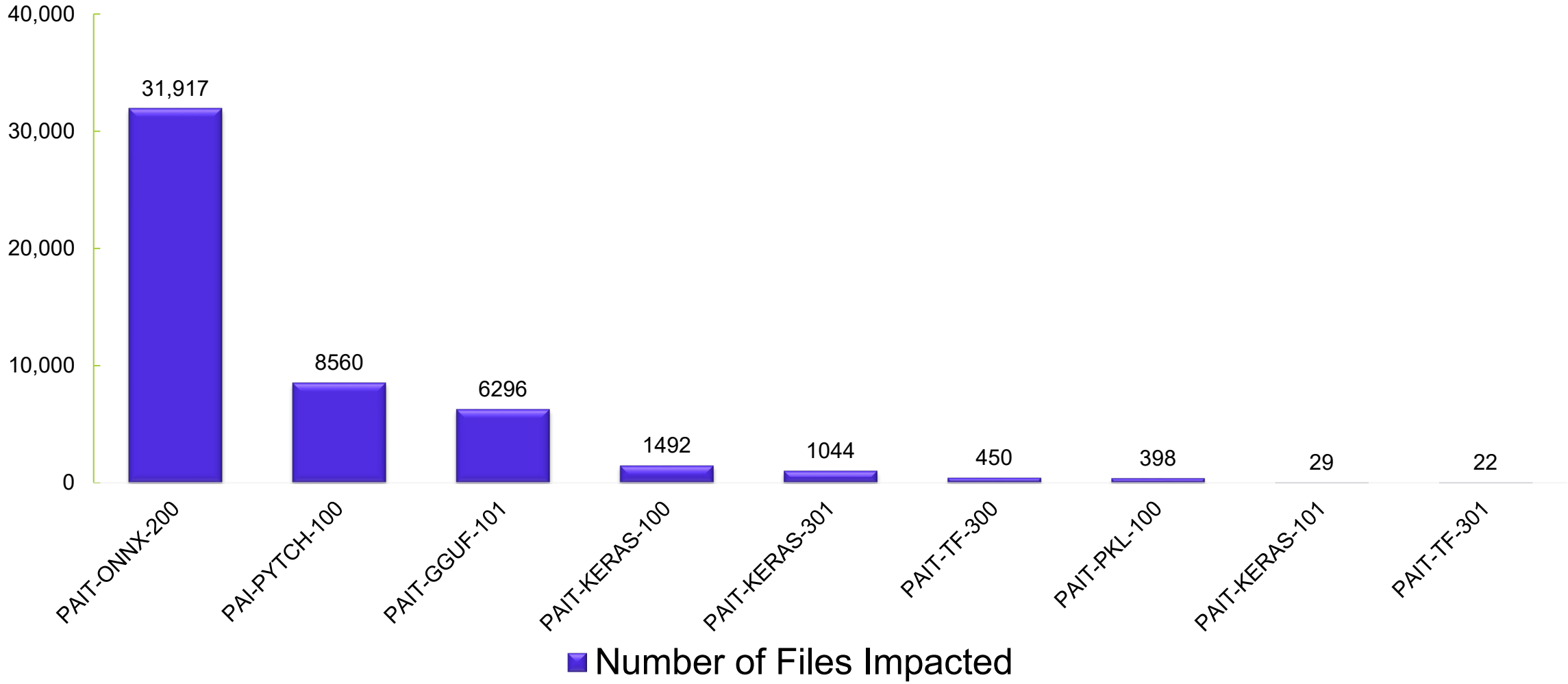


Taxonomizing AI Threats

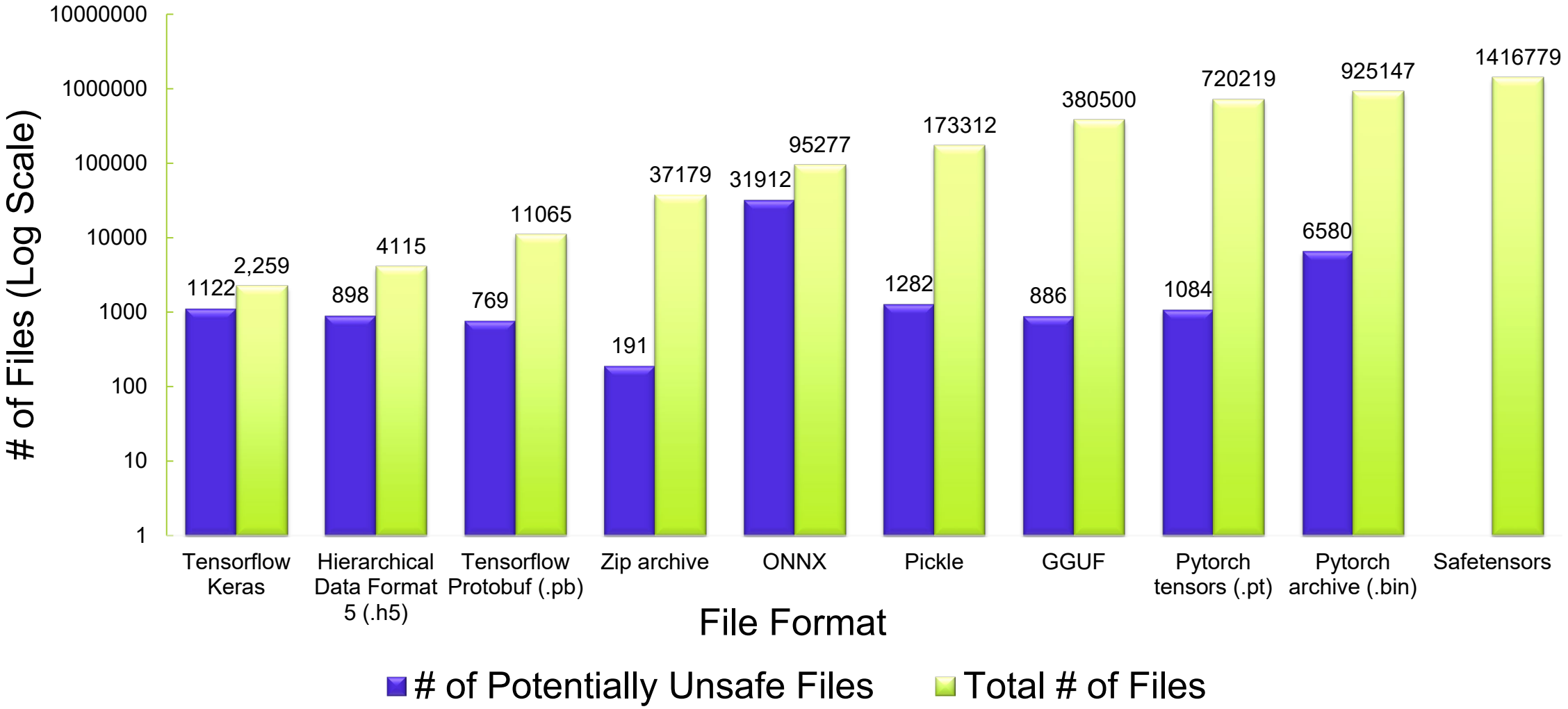


- **PAIT-GGUF-101**: GGUF Model Template Containing Arbitrary Code Execution Detected
- **PAIT-KERAS-100**: Keras Model Lambda Layer Can Execute Code At Load Time
- **PAIT-KERAS-101**: Keras Model Lambda Layer Arbitrary Code Execution Detected at Model Load Time
- **PAIT-PKL-100**: Pickle Model Arbitrary Code Execution Detected at Model Load Time
- **PAIT-PYTCH-100**: PyTorch Model Arbitrary Code Execution Detected at Model Load Time
- **PAIT-ONNX-200**: ONNX Model Contains Architectural Backdoor
- **PAIT-KERAS-301**: Keras Model Lambda Layer Arbitrary Code Execution Detected at Model Run Time
- **PAIT-TF-300**: TensorFlow SavedModel Contains Unknown Operators
- **PAIT-TF-301**: TensorFlow SavedModel Contains Arbitrary Code Execution at Model Run Time

Types of Threats Found



Threats by File Format



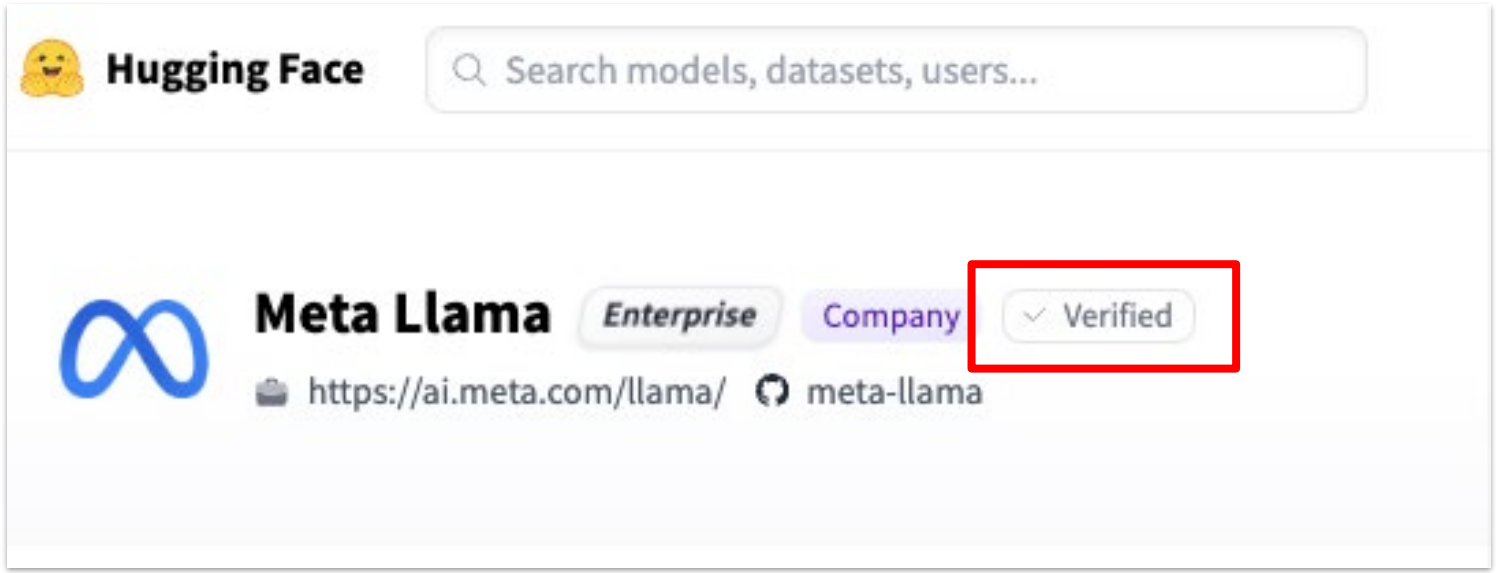
Is ONNX unsafe?

- PAIT-ONNX-200 was the most common threat, but ONNX is not a very common format
- Random sampling determined many were false positives
- Added additional filters for valid, non-backdoor architectures
- Reduced daily findings by 95% or more in initial testing

Image: stock.adobe.com

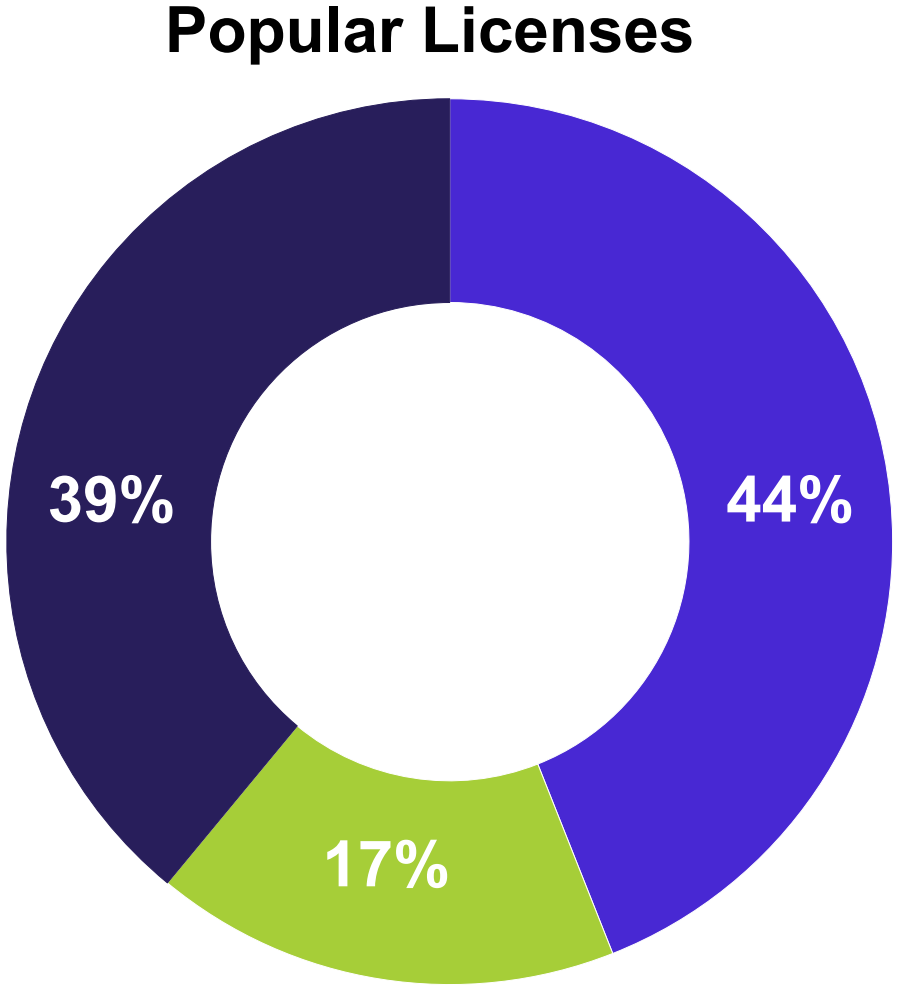
Name Squatting:
Old Wine in a New Bottle

- Name squatting originally evolved in OSS ecosystems
- Easiest way to “hijack” trust of known organizations
- Never trust, always verify



Licensing Wild West

- <10% of models have a license
- Only ⅓ of licensed models are truly “open source”
- “Openwashing” is common



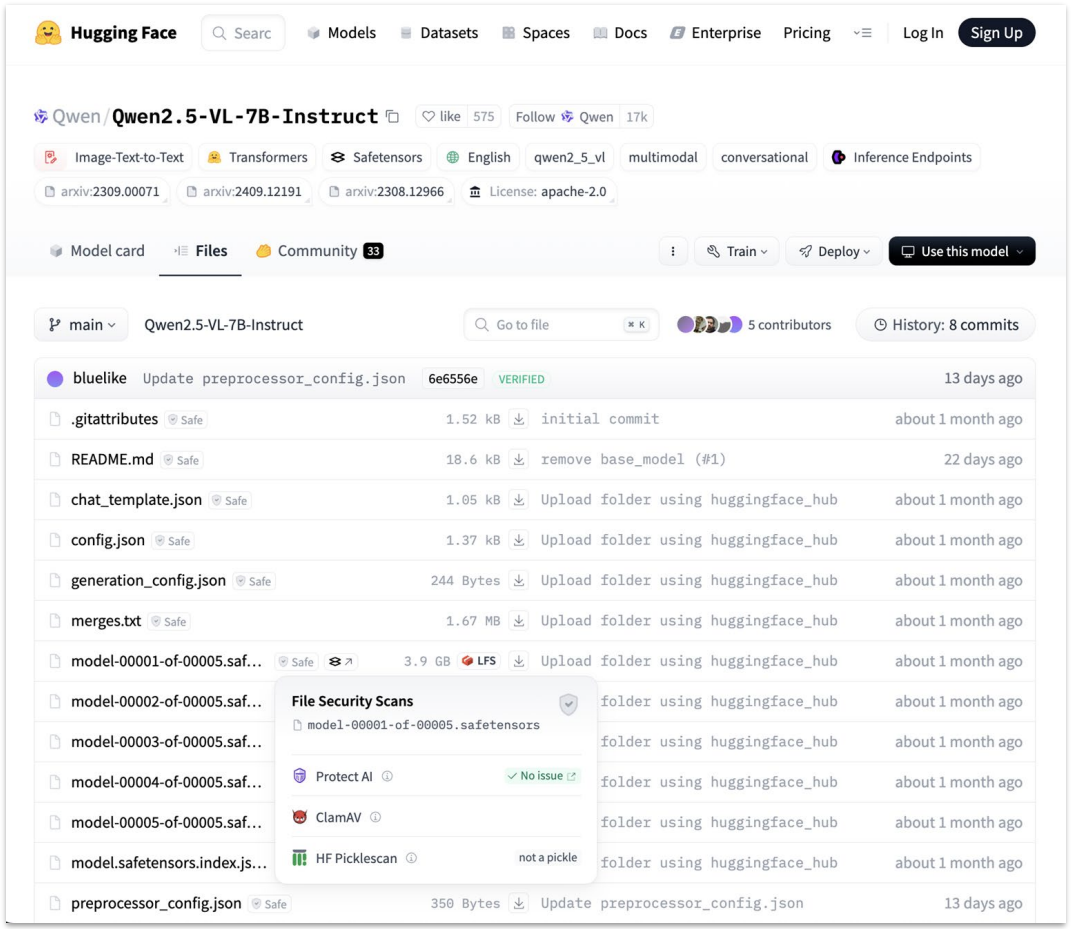
RSAC 2025 Conference

Many Voices.
One Community.

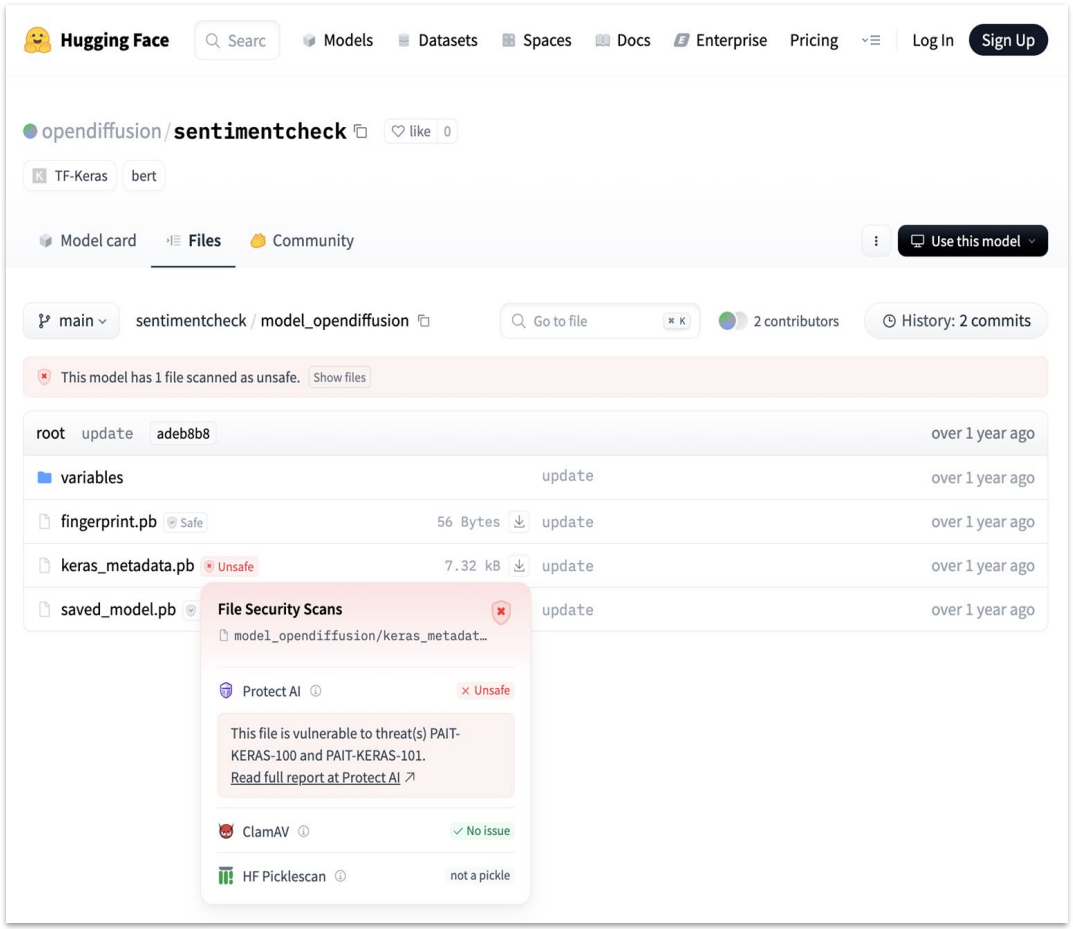
Sharing Results with the Community

Tying it all together

SAFE

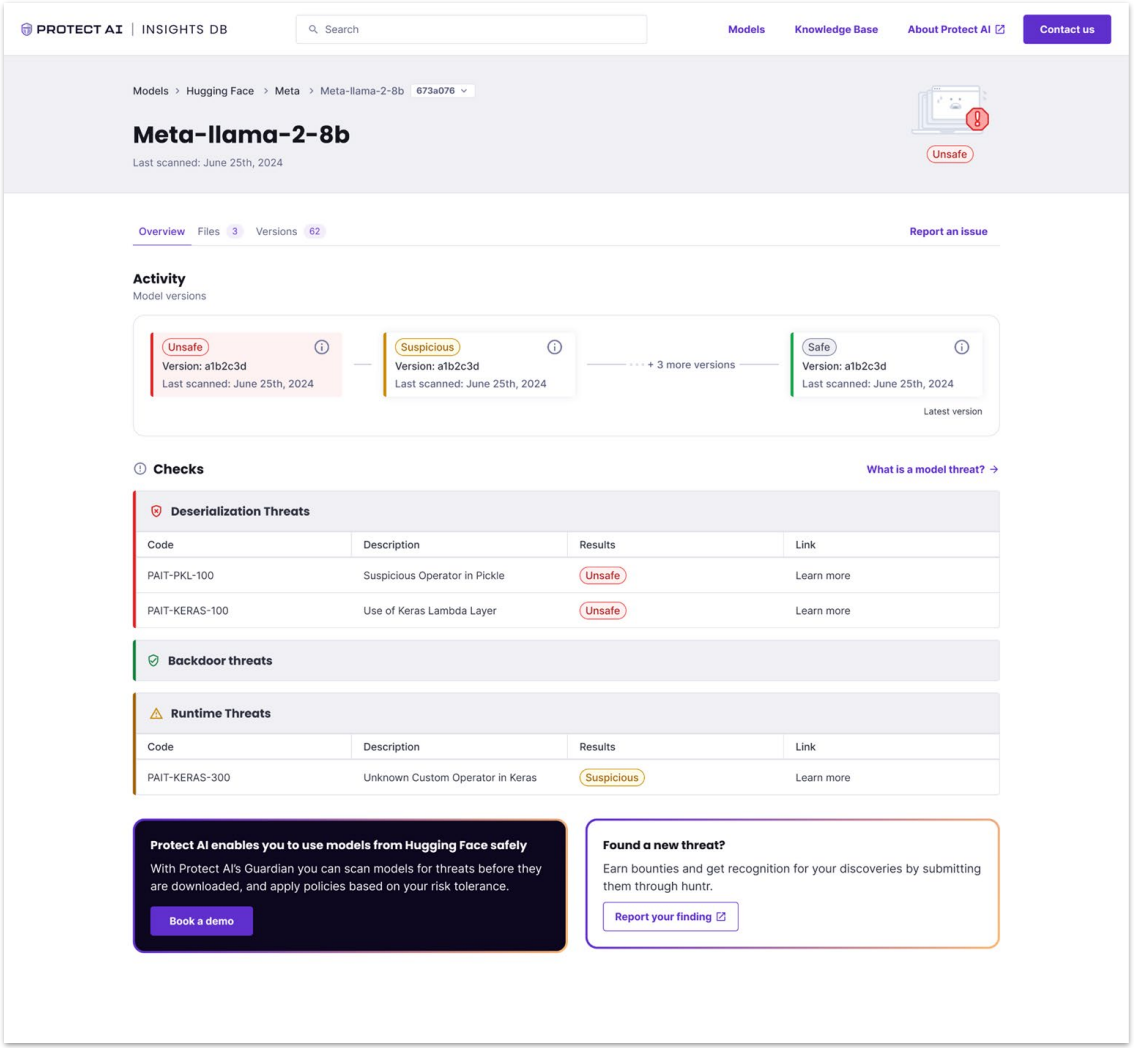


UNSAFE



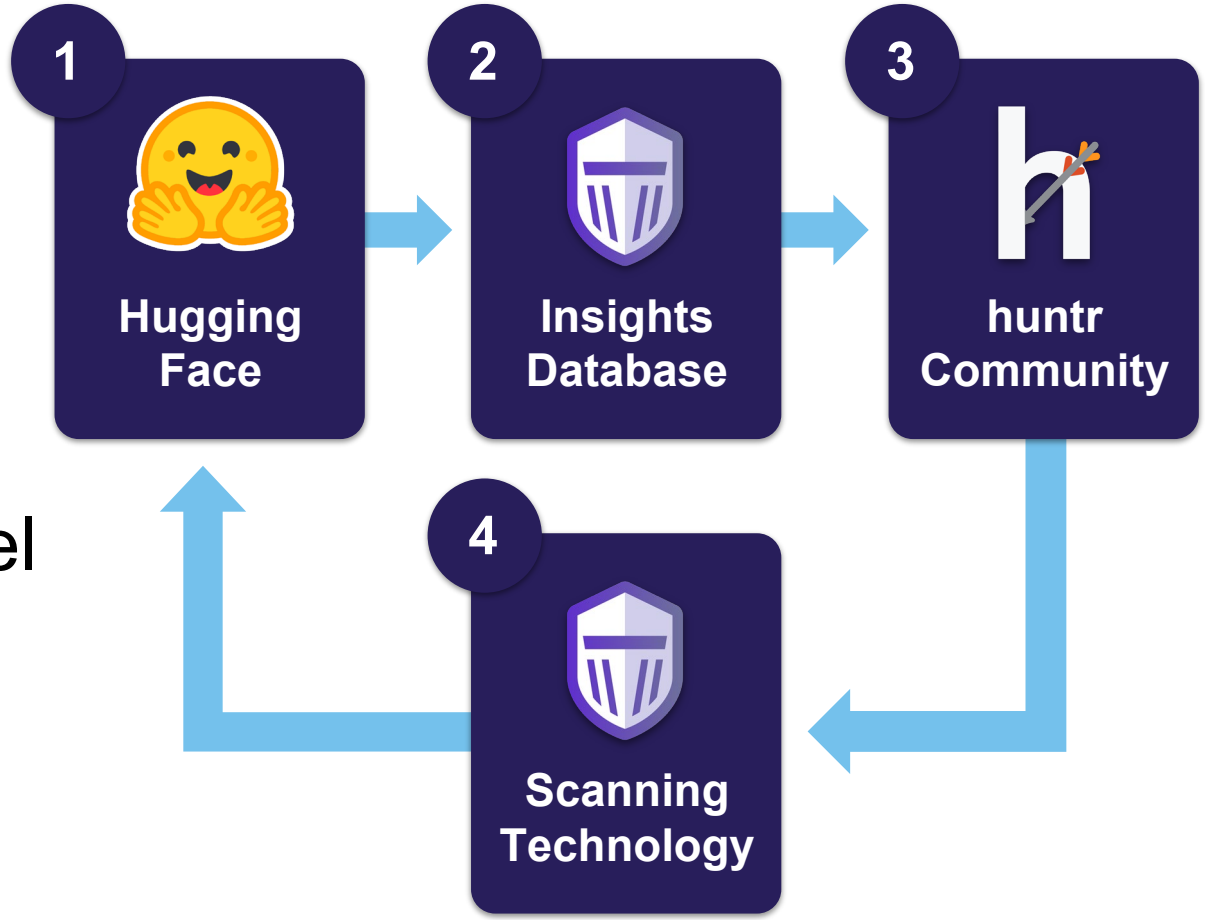
What is Insights DB?

- A public, authoritative source for open source AI threats
- Threat taxonomy from the latest Protect AI research
- In-depth scan reports of every model on Hugging Face
- Open to community feedback via Huntr bug bounty program

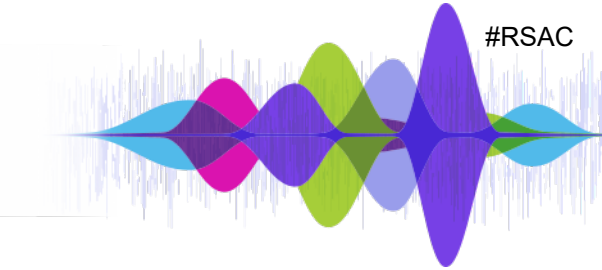


A Virtuous Cycle of Open Source AI

1. Hugging Face models are scanned
2. Insights DB stores all public scan results
3. Huntr community validates scans and discovers new model vulnerabilities
4. Research findings from Huntr improve next iteration of scans



AI Security in Action!



- mcpotato/42-eicar-street
- deepseek-ai/DeepSeek-R1
- nicoboss/DeepSeek-R1-Distill-Qwen-1.5B-Fully-Uncensored-Lora

Apply What You Have Learned Today

- In the next week you should:
  - Check out Insights DB to learn more about these threats
  - Collaborate with Data Science teams to identify models in use at your organization and establish documentation for them in model cards
  - Check Insights DB to see if those models might be at risk
- In the following months you should:
  - Implement policies requiring a clean scan when downloading any public models from Hugging Face
  - Look into model scanners for any private models – OSS ModelScan or more advanced options

Insights DB

RSAC 2025 Conference

Many Voices.
One Community.

Thank You!